University of Antwerp
Faculty of Science

Graph-Based Game Recommendations Using Steam User-Item Interactions

Pablo Deputter

Research Questions

- **How do graph-based recommendation approaches compare to traditional algorithms in improving recommendation accuracy on the Steam platform, and what are the associated computational trade-offs?**
  - *Hypothesis:* Graph-based methods significantly outperform traditional algorithms in recommendation accuracy by capturing complex user-item interactions, albeit with increased computational requirements
- **How does incorporating Personalized PageRank (PPR) enhance recommendation accuracy in game recommendation systems on the Steam dataset?**
  - *Hypothesis:* Personalized PageRank leverages the underlying graph structure of user-item interactions to provide more relevant and accurate recommendations, thereby increasing user satisfaction
- **(How does multi-hop message passing in Graph Neural Networks (GNNs) influence the accuracy and effectiveness of game recommendations on the Steam platform?**
  - *Hypothesis:* Multi-hop message passing enables GNNs to capture higher-order relationships within the user-item graph, leading to improved recommendation accuracy and better handling of complex interaction patterns)

Stakeholders

- Primary Stakeholder: **Users** or **Gamers**
  - Enhance user experience by providing accurate and personalized game recommendations

- Secondary Stakeholder: **Game Developers/Publishers**
  - Support publishers by connecting their games with the right audience

Metrics

- Focus on accuracy!
- **NDCG@20**
- **Recall@20**
- **Execution time (s)**:
  - Model training + inference (generating recommendations)
  - Are complex models worth it?

Offline Evaluation Setup

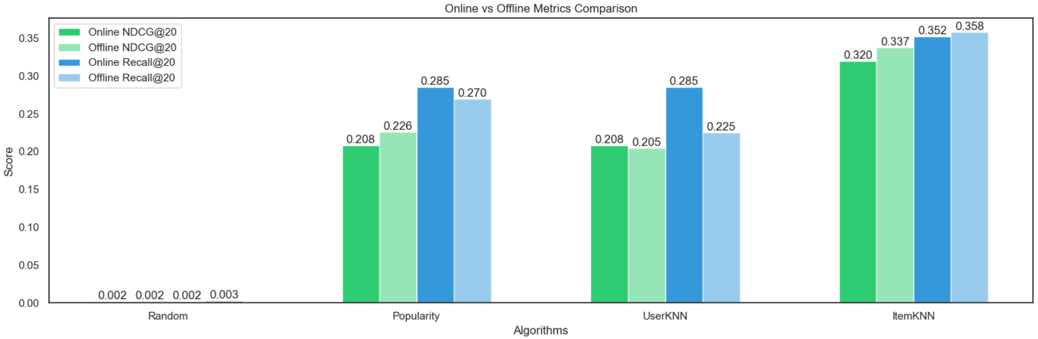
- Resource constraints (limited computational power and time)
- Need for efficient evaluation!
- **Sample-Based Evaluation**:
  - Utilize a representative subset of data
  - Ensure reliability despite reduced data size
- **Advantages**:
  - Quick testing of different algorithms
  - Reduced computational resources requirement
  - Ability to explore broader parameter space
- **Disadvantages**:
  - Potential risk of missing patterns present in full dataset
  - Possible suboptimal parameter choices

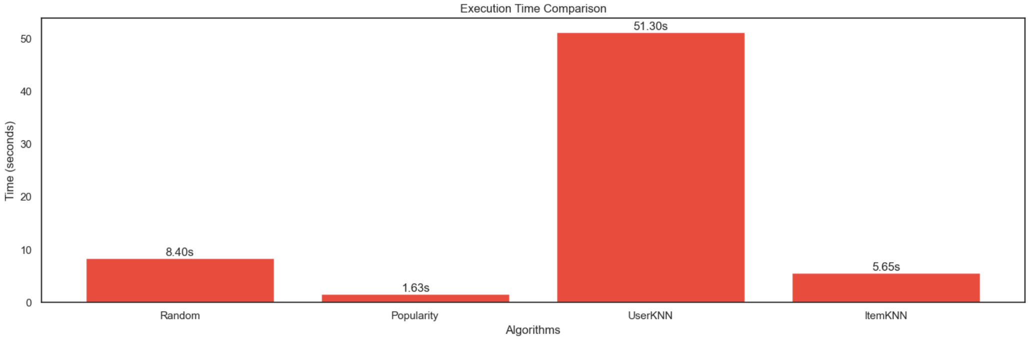
Offline Evaluation Setup

- Choose **sample size** (number of users)
- Maintain user activity and item popularity distributions
- Data is split in training, validation and test set
  - No overlap across sets to prevent data leakage
- Cross-validation setup
  - Ensure consistent splits for all algorithms
  - Multiple random sees to ensure robustness
- Fair comparison between algorithms and hyperparameter tuning
  - Same sampled data and splits
  - Only optimization path differs

Baselines

- Offline Evaluation:
  - 3-Fold validation on full training set
  - Metrics computed on validation set
- Online Evaluation:
  - Models trained on full training set
  - Evaluated on new users (Cold-Start)
- **Random (0.002)**
- **Popularity (0.208)**
  - Suprisingly effective
- **User-KNN (0.208)**
  - Cosine similarity between users
  - Fallback to popularity for Cold-Start
- **Item-KNN (0.320)**





Personalized PageRank - Introduction

- Originally designed by Google to rank web pages
  - High number of incoming links signifies importance
  - Outgoing links distribute this importance
- **PPR** adapts this by considering user's interaction history → Personalized recommendations
- Natural fit for data
  - Sparse interaction data
  - Cold-start problems
  - Playtime integration
  - Balances popularity with personalization

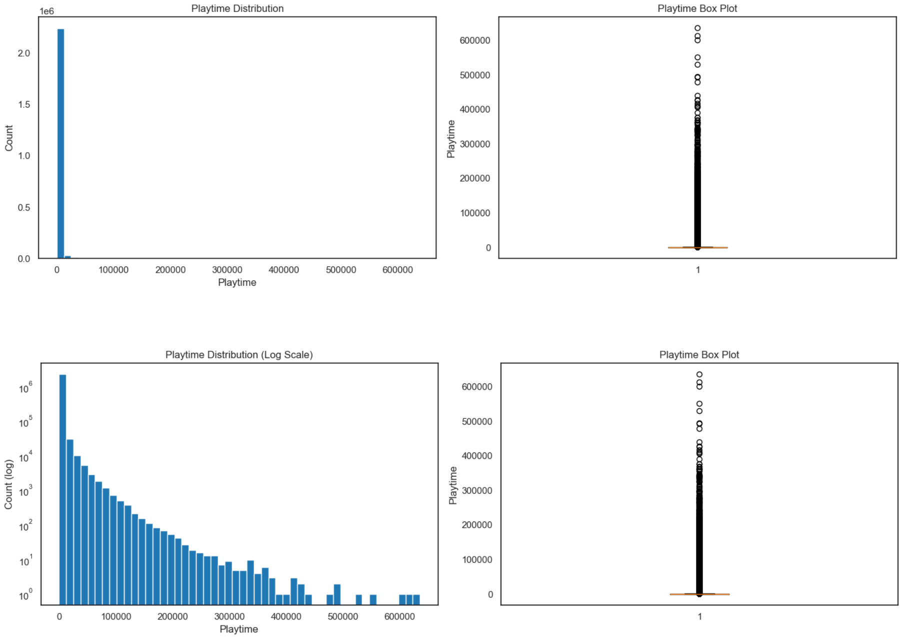
Personalized PageRank – Parameters

- **α (Alpha)**:
  - Damping factor, probability of continuing a random walk
- **num_iterations**:
  - Maximum iterations for the power method
- **popularity_weight**:
  - Balances item popularity in recommendations
- **process_playtime**:
  - How playtime data is processed

Note: in literature alpha is switched around depending on the source.

Personalized PageRank – Data Processing

- **Log transformation** to reduce outlier impact
  - Normalization using MinMaxScaler
- **Relative scaling** based on user's average playtime



University of Antwerp
Faculty of Science

Personalized PageRank – Example

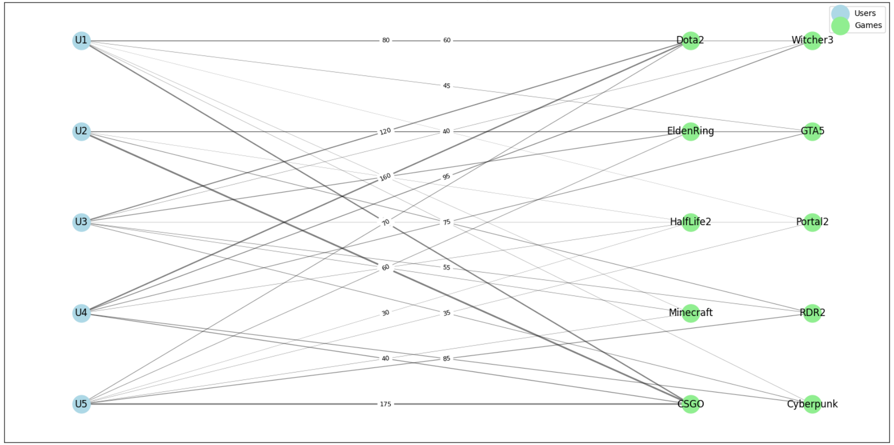
- **Users:** U1, U2, U3, U4 and U5
- **Games:** CSGO, Dota 2, Elden Ring, Minecraft, GTA V, Red Dead Redemption 2, The Witcher 3, Portal 2, Half-Life 2 and Cyperpunk 2077

$$W = \begin{bmatrix} 150 & 80 & 0 & 30 & 45 & 0 & 60 & 20 & 0 & 35 \\ 200 & 0 & 40 & 0 & 90 & 70 & 0 & 0 & 25 & 0 \\ 0 & 120 & 85 & 50 & 0 & 55 & 40 & 30 & 0 & 65 \\ 90 & 160 & 0 & 0 & 75 & 0 & 95 & 0 & 40 & 80 \\ 175 & 70 & 60 & 40 & 0 & 85 & 0 & 35 & 30 & 0 \end{bmatrix}$$

$$\Downarrow$$

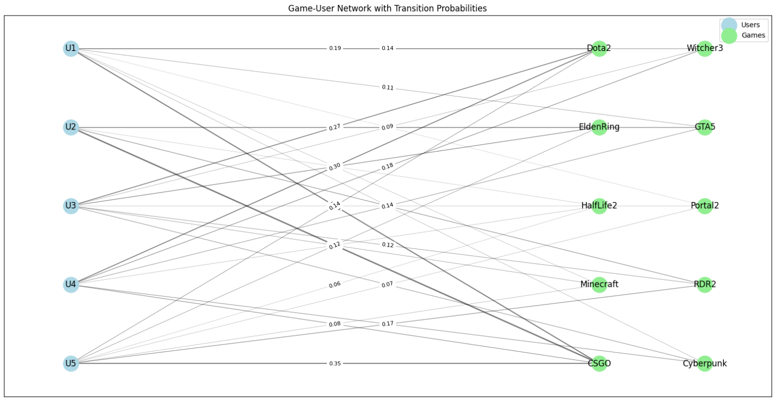
$$W' = \begin{bmatrix} 0.87 & 0.60 & 0 & 0.17 & 0.35 & 0 & 0.47 & 0 & 0 & 0.24 \\ 1.00 & 0 & 0.30 & 0 & 0.65 & 0.54 & 0 & 0 & 0.10 & 0 \\ 0 & 0.78 & 0.63 & 0.40 & 0 & 0.44 & 0.30 & 0.17 & 0 & 0.51 \\ 0.65 & 0.90 & 0 & 0 & 0.57 & 0 & 0.67 & 0 & 0.30 & 0.60 \\ 0.94 & 0.54 & 0.47 & 0.30 & 0 & 0.63 & 0 & 0.24 & 0.17 & 0 \end{bmatrix}$$

Game-User Network with Playtime Hours



Personalized PageRank - Components

- **Interaction (User-Item) Matrix**:
  - Sparse matrix with normalized playtime
  - Rows sum to 1 representing transition probability from user → item
- **Item-User Matrix**:
  - Transposed User-Item matrix
  - Columns sum to 1 representing probability from item → user



$$P_{UI} = \begin{bmatrix} 0.32 & 0.22 & 0 & 0.06 & 0.13 & 0 & 0.17 & 0 & 0 & 0.09 \\ 0.39 & 0 & 0.12 & 0 & 0.25 & 0.21 & 0 & 0 & 0.04 & 0 \\ 0 & 0.24 & 0.19 & 0.12 & 0 & 0.14 & 0.09 & 0.05 & 0 & 0.16 \\ 0.18 & 0.24 & 0 & 0 & 0.15 & 0 & 0.18 & 0 & 0.08 & 0.16 \\ 0.29 & 0.16 & 0.14 & 0.09 & 0 & 0.19 & 0 & 0.07 & 0.05 & 0 \end{bmatrix} \qquad P_{IU}(j,i) = \frac{w'_{ij}}{\sum_k w'_{kj}}$$

$$P_{IU} = \begin{bmatrix} 0.25 & 0.21 & 0 & 0.20 & 0.22 & 0 & 0.33 & 0 & 0 & 0.18 \\ 0.29 & 0 & 0.21 & 0 & 0.41 & 0.34 & 0 & 0 & 0.17 & 0 \\ 0 & 0.28 & 0.45 & 0.46 & 0 & 0.27 & 0.21 & 0.41 & 0 & 0.38 \\ 0.19 & 0.32 & 0 & 0 & 0.37 & 0 & 0.46 & 0 & 0.53 & 0.44 \\ 0.27 & 0.19 & 0.34 & 0.34 & 0 & 0.39 & 0 & 0.59 & 0.30 & 0 \end{bmatrix} \qquad P_{UI}(i,j) = \frac{w'_{ij}}{\sum_k w'_{ik}}$$

Personalized PageRank - Components

- **Item Popularity**
  - Log transformation followed by normalization
  - Each item is assigned a popularity score
  - Provides a fallback for users with short history
- **Personalization Vector**
  - Initialized based on user's seed items
  - Adjusted for users with varying interaction histories

$$W' = \begin{bmatrix} 0.87 & 0.60 & 0 & 0.17 & 0.35 & 0 & 0.47 & 0 & 0 & 0.24 \\ 1.00 & 0 & 0.30 & 0 & 0.65 & 0.54 & 0 & 0 & 0.10 & 0 \\ 0 & 0.78 & 0.63 & 0.40 & 0 & 0.44 & 0.30 & 0.17 & 0 & 0.51 \\ 0.65 & 0.90 & 0 & 0 & 0.57 & 0 & 0.67 & 0 & 0.30 & 0.60 \\ 0.94 & 0.54 & 0.47 & 0.30 & 0 & 0.63 & 0 & 0.24 & 0.17 & 0 \end{bmatrix}$$

$$\mathrm{pop}_j = \sum_i w'_{ij}$$

$$\mathrm{pop}'_j = \log(1 + \mathrm{pop}_j)$$

$$\mathrm{pop}''_j = \frac{\mathrm{pop}'_j}{\sum_k \mathrm{pop}'_k}$$

Game	Raw Sum	Log(1+sum)	Normalized
CSGO	3.46	1.50	0.15
Dota 2	2.82	1.34	0.13
Elden Ring	1.40	0.88	0.09
Minecraft	0.87	0.63	0.06
GTA V	1.57	0.95	0.09
RDR2	1.61	0.96	0.10
Witcher 3	1.44	0.89	0.09
Portal 2	0.41	0.34	0.03
Half-Life 2	0.57	0.45	0.04
Cyberpunk 2077	1.35	0.85	0.08

Personalized PageRank – Personalization Vector

- **Consider a new user**
  - Elden Ring - 45 hours
  - The Witcher 3 - 30 hours
  - GTA V - 15 hours
- Playtime Processing
- Initial Personalization Vector

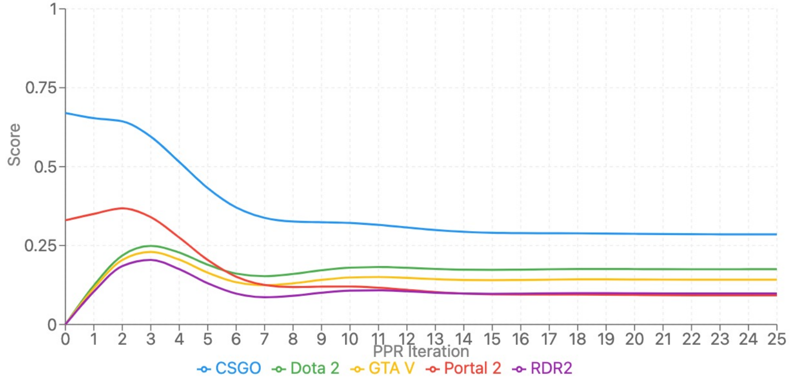
$$t = \begin{bmatrix} 45 \\ 30 \\ 15 \end{bmatrix} \qquad t_{log} = \log(1+t) = \begin{bmatrix} 3.83 \\ 3.43 \\ 2.77 \end{bmatrix}$$

$$v_j = \begin{cases} t_{norm,j} & \text{if game j is in history} \\ 0 & \text{otherwise} \end{cases}$$

$$v = \begin{bmatrix} 0 & 0 & 0.38 & 0 & 0.28 & 0 & 0.34 & 0 & 0 & 0 \end{bmatrix}^T$$

Personalized PageRank - Components

- **Power Iteration Process**:
  - Iteratively propagate personalization vectors
  - $R_{next} = \alpha(R \cdot P_{IU} \cdot P_{UI}) + (1-\alpha)v_p$
  - Stop when convergence criteria hits or after **num_iterations**
- **Adaptive Popularity Weights**:
  - Adjust based on user history length
  - Cold-Start users → Rely entirely on item popularity
  - Short history → Higher weight on popularity
  - Long History → Base popularity gets maintained
  - For history of 3 games:
    - $\omega(3) = \min(0.5, \omega_b + (5-3) \cdot 0.1 = 0.4$



$$\omega(h) = \begin{cases} 0.9 & \text{if } h = 0 \text{ (no history)} \\ \min(0.5, \omega_b + (5-h) \times 0.1) & \text{if } 1 \le h \le 5 \\ \omega_b & \text{if } h > 5 \end{cases}$$

Personalized PageRank - Recommendations

- **Final Scores**:
  - Weighted combination of PPR and item popularity
  - $score_j = (1 - \omega(h))R_{final,j} + \omega(h) \cdot pop''_j$
  - $score_j = (1 - 0.4)R_{final,j} + 0.4 \cdot pop''_j$
- **Recommendations**:
  - Exclude known interactions, zero out
  - Select top-n items

$$score_{final} = \begin{bmatrix} \text{CSGO: } 0.16 \\ \text{Dota 2: } 0.16 \\ \text{Elden Ring: } \cancel{0.24} \\ \text{Minecraft: } 0.07 \\ \text{GTA V: } \cancel{0.20} \\ \text{RDR2: } 0.17 \\ \text{Witcher 3: } \cancel{0.23} \\ \text{Portal 2: } 0.08 \\ \text{Half-Life 2: } 0.09 \\ \text{Cyberpunk: } 0.15 \end{bmatrix}$$

MultiAlphaPPR

- Extends PPR by utilizing multiple damping factors
- Combines multiple weighted PPR scores
- Additional Parameters:
  - **Alphas**:
    - List of damping factors
  - **Alpha Weights**:
    - Corresponding weights for each alpha, summing up to 1
- **MultipleRandom Walks**: run PPR independently for each alpha
- **Weighted Combination**: aggregate PPR scores using alpha weights
- **Final Scoring**: Combine aggregated PPR with item popularity

Note: in literature alpha is switched around depending on the source.

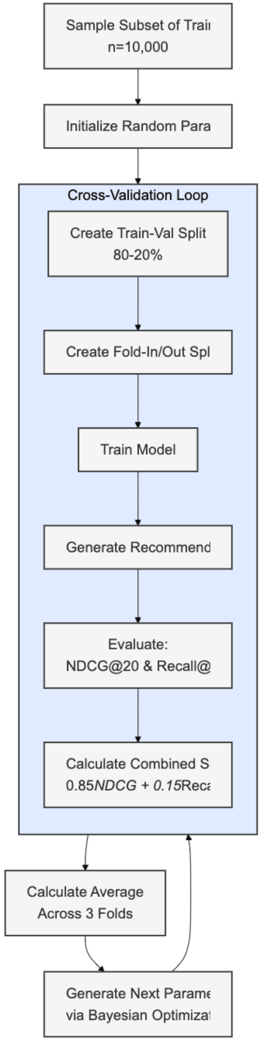
TwoStagePPR

- Enhances PPR with two-phase approach
- Combines broad exploration and focused exploitation
- Additional Parameters:
  - **Stage 1 Alpha ($\boldsymbol{\alpha_1}$)**:
    - Higher damping factor for broad exploration (e.g., 0.7)
  - **Stage 2 Alpha ($\boldsymbol{\alpha_2}$)**:
    - Lower damping factor for focused exploitation (e.g., 0.3)
  - **Stage 1 Top-K Items**:
    - Number of top items to consider from first stage
- Initialize personalization with top-K items
  - Run PPR with α_2 to refined recommendations
- Final scoring averages PPR from both stages

Note: in literature alpha is switched around depending on the source.

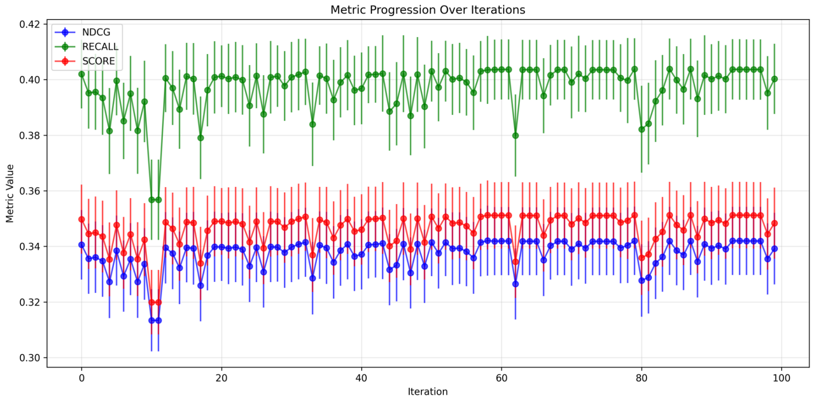
Optimization

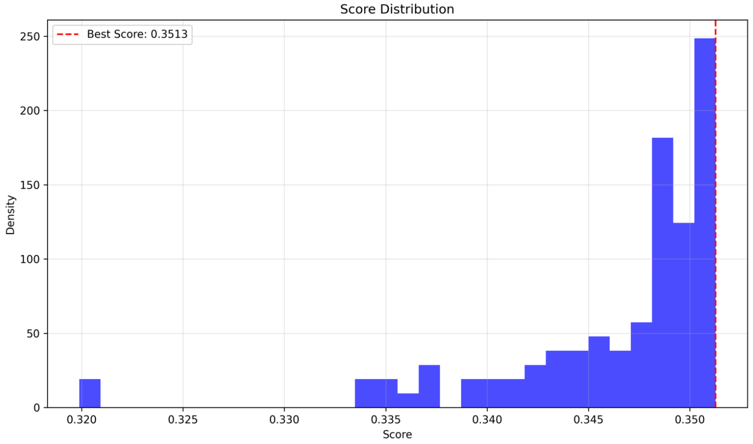
- **Bayesian Optimization + Sampling**:
  - Efficient exploration of hyperparameter space
- Evaluation using weighted combination of NDCG@20 & Recall@20
  - 0.85NDCG@20 + 0.15Recall@20
- **Process**:
  - Sample dataset while maintaining distributional characteristics
  - Cross-Validation trials on different seeds:
    - Base seed calculates: sampling, cv and optimization seed
  - Model training & evaluation
  - Iterate to find optimal hyperparameters
- **Robustness**:
  - Different algorithms make use of same base seed
  - Same data split across trials



Personalized PageRank - Results

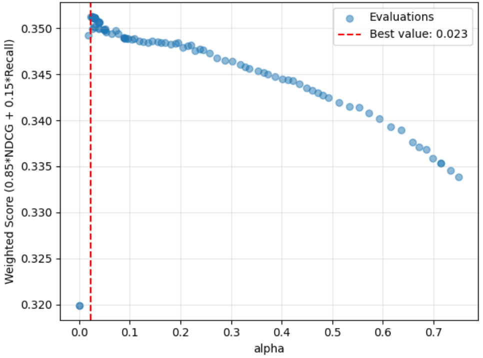
- **Optimized Parameters:**
  - Alpha: 0.023
  - Popularity Weight: 0.033
  - num_iterations: 93
  - Playtime processing: Log transformation followed by MinMaxScaler
- **Performance**:
  - Offline: NDCG@20 = 0.3449, Recall@20 = 0.4172
  - Offline: NDCG@20 = 0.3183, Recall@20 = 0.3790
  - Execution Time: 280s (CPU), 30-45s (GPU)
- Outperform all baselines except Item-KNN
- Comparable to baseline execution times

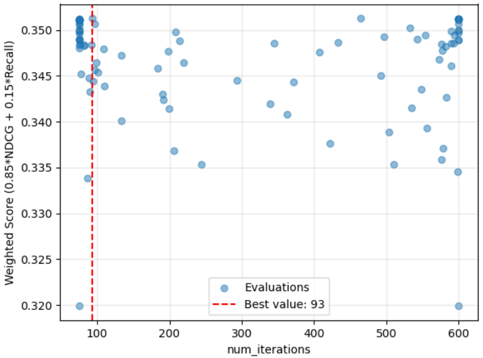


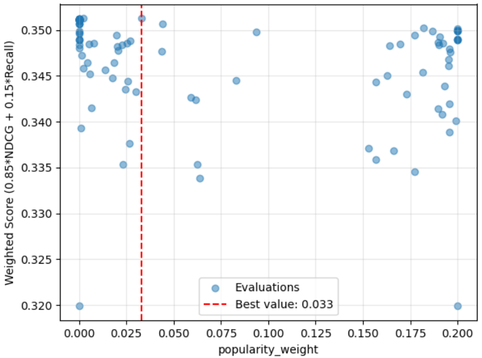


Personalized PageRank - Results

- **Low Alpha Implications**:
  - Minimal reliance on random walk
  - Greater influence of personalization vector
  - Potential underutilization of graph structure
- **Popularity Weight Near Zero**:
  - Emphasizes personalized scores over global popularity

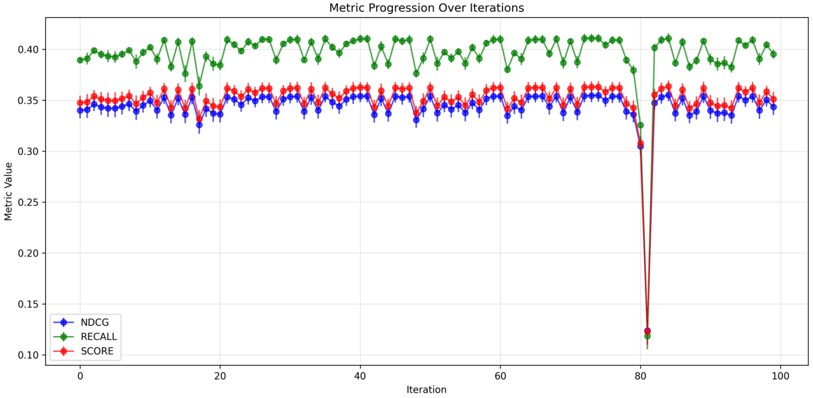


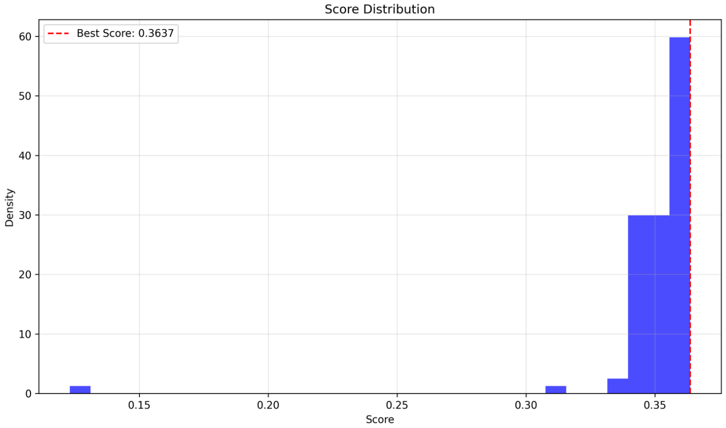




TwoPhasePPR - Results

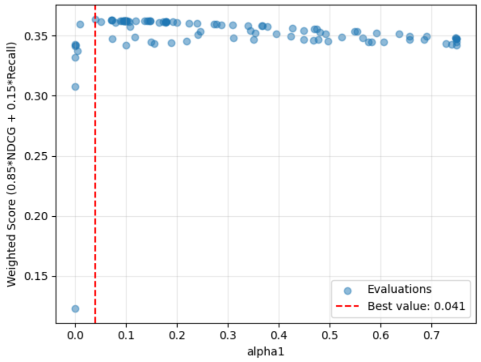
- **Optimized Parameters:**
  - Alpha1: 0.041
  - Alpha2: 0.0
  - Stage1_K: 148
  - Popularity Weight: 0.0
  - num_iterations: 271
- **Performance**:
  - Offline: NDCG@20 = 0.3453, Recall@20 = 0.4175
  - Offline: NDCG@20 = 0.3181, Recall@20 = 0.3790
  - Execution Time: 350 (CPU), 40-55s (GPU)
- Gives almost the exact same results
- Slightly longer than PPR due to additional phase (that actually has no influence...)

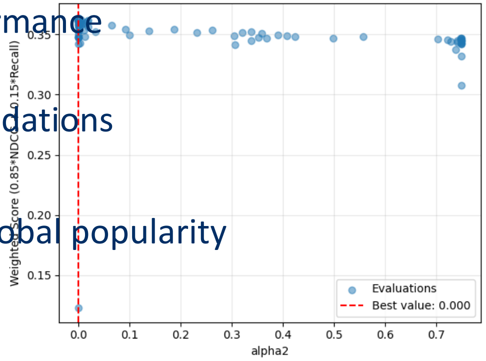


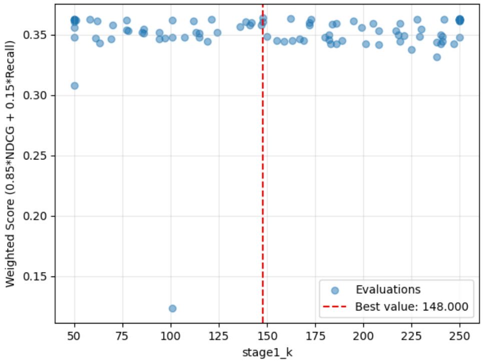


TwhoPhasePPR - Results

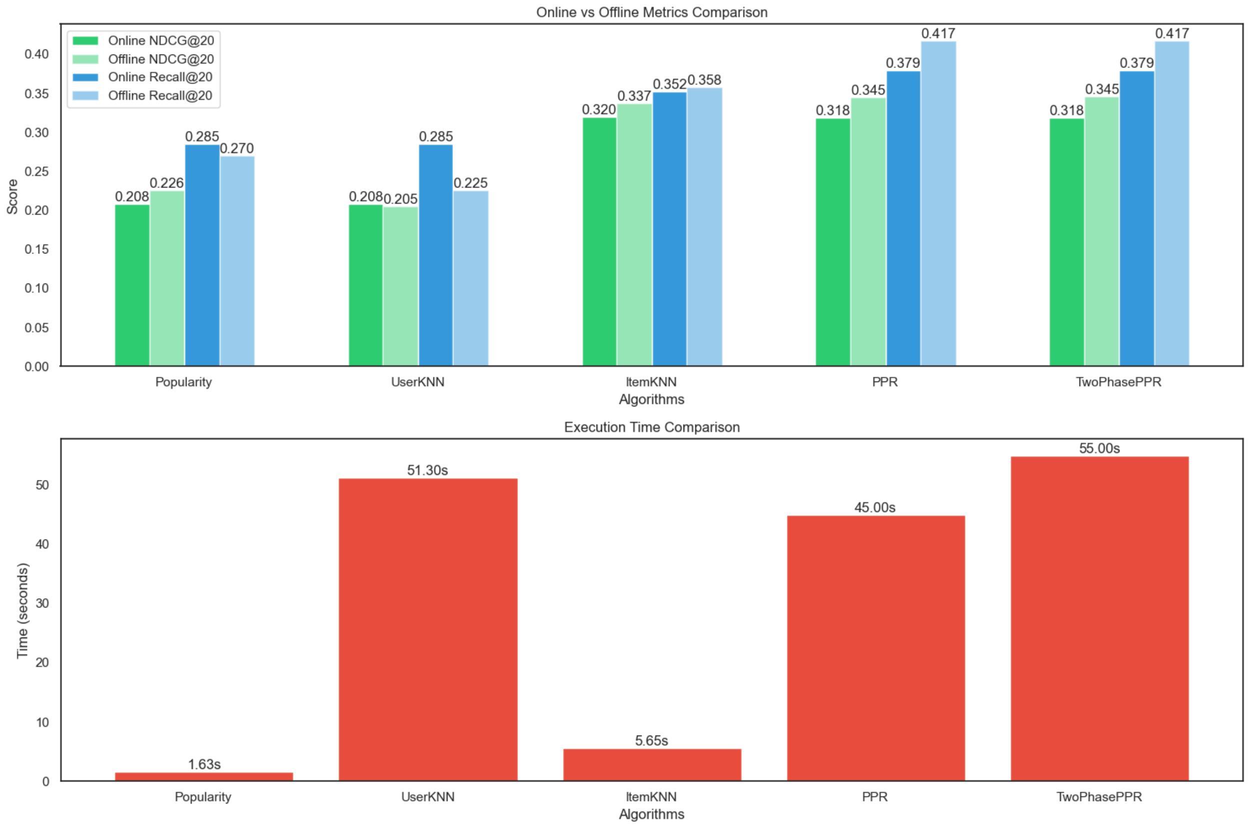
- **Alpha2 at zero**:
  - Second phase no influence, effectively reducing to single-phase PPR
  - No contribution → redundant
  - No additional performance gains observed from 2nd phase
  - Extra complexity != extra performance
- **Stage1_K Impact**:
  - Moderate variation, no signicant performance difference across different K values
  - Suggests minimal impact on recommendations
- **Popularity Weight near zero**:
  - Emphasizes personalized scores over global popularity







Dicussion – Baseline Comparison



University of Antwerp
Faculty of Science

Discussion – Alpha Insights

- **Effect of Low Alpha**:
  - Dominance of personalization vector over graph exploration
  - Minimal influence of the link structure
  - Underutilization of graph data
- **Alpha = 0:**
  - Comparable to Popularity baseline
  - No graph propagation leading to popularity-based suggestions
- **Small Alpha vs. Zero Alpha**:
  - Slight influence of graph structure can still enhance recommendations
  - Recomendations degrade to user history or popularity (no graph propagation), leading to popularity-based suggestions
- **MultiAlpha**:
  - Low alphas dominate, rendering multiple alphas ineffective
  - Leads to redundant computations

Dicussion - Why GNNs where considered

- **Strengths**:
  - Capable of capturing complex user-item relationships through multi-hop message passing
  - Enhanced representation learning by aggregating informations from neighbours
- **Challenges**:
  - GNNs like LightGCN rely on pre-learned user embeddings from user data
  - Unable to generate embeddings for new users without retraining
  - Significant computational resources needed for large-scale graphs due to multi-hop message passing
  - Modified GNNs did not outperform simpler methods like PPR and Item-KNN in small-scale testing
- ➔ Discontinue GNN research

Future Work & Improvements

- **Modified GNN with Item Co-occurence**:
  - Create co-occurence graph based on shared user interactions
  - Capture higher-order relation ships
  - Achieved similar performance to PPR and Item-KNN in small-scale testing
  - Remains highly resource-intensive (4-5 hours of training with graph reduction techniques
- **Hybridizing PPR with Content-Based Features**:
  - Merge PPR with content-based filtering to leverage both interaction data and rich item attributes
- **Promoting Diversity & Fairness**:
  - Varied set of recommendations to avoid over-reliance on popular items